DSL
Distributed
Systems
Laboratory

ATC

Onboard Processing using the Adaptive Network Architecture (ANA)

Dipa Suri and Adam Howell
Advanced Technology Center
Lockheed Martin Space Systems Company
Palo Alto, CA

Nishanth Shankaran,
John Kinnebrew, Will Otte,
Doug Schmidt and Gautam Biswas
Institute for Software Integrated Systems (ISIS)
Vanderbilt University
Nashville, TN

Supported under NASA AIST Contract # NAS2-03148

June 28, 2006

DSL
Distributed
Systems
Laboratory

Overview

- Motivation and Project Goals
- Adaptive Network Architecture Overview
  - Agent Hierarchy
  - CORBA Component Model (CCM) Implementation
- Onboard Processing via the Science Agent
  - Spreading Activation Partial Order Planner (SA-POP)
  - Resource Allocation and Control Engine (RACE)
- Example application to Magnetospheric Multi-Scale (MMS) Mission
- Conclusions and Future Work
- Acronym List

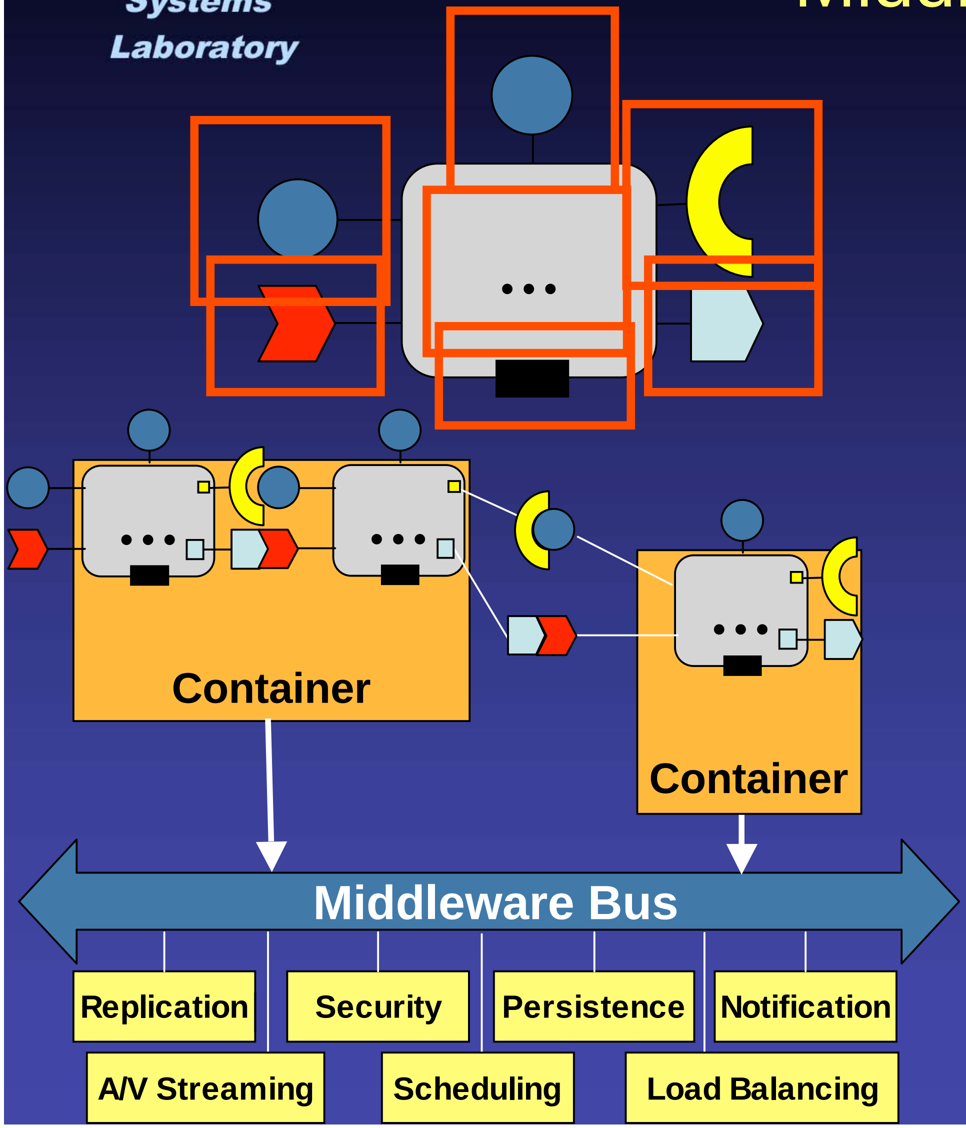
Motivation

- Future space missions will rely on constellations of spacecraft with heterogeneous sensor/instrument suites to *cooperatively* meet their mission objectives (i.e. Constellation, and the NASA Sensor Web)
- Traditional stovepipe operations model has limitations
  - Increased complexity associated with managing multi-spacecraft missions from ground stations
  - Limited reactivity for missions requiring opportunistic science
- One way to address these issues is by increasing the amount of onboard data processing and autonomy, i.e.
  - Sensor and computing resource management
  - Scheduling, execution, and monitoring of activities/tasks.

Project Goals

- Develop an *agent-based* software architecture, called the Adaptive Network Architecture (ANA) to support autonomy for science missions comprised of multiple, heterogeneous, distributed assets
  - Software Agents have many different definitions, but common characteristics include *Communication*, *Collaboration*, and *Autonomy*
  - ANA is composed of multiple agents that act as an ensemble to manage both the payload and spacecraft bus
- A key feature of the ANA is to provide a framework for onboard science processing that can be dynamically reconfigured and/or distributed across multiple processors to meet resource and QoS constraints
  - Leverage mature terrestrial standards and software best-practices
  - ANA is built on CORBA Component middleware (CCM) specifically designed for distributed realtime embedded systems (CIAO and DaNCE)

Birdseye View of Component Middleware



- *Components* encapsulate application “business” logic
- Components interact via *ports*
  - *Provided interfaces*, e.g.,facets
  - *Required connection points*, e.g., receptacles
  - *Event sinks & sources*
  - *Attributes*
- *Containers* provide execution environment for components with common operating requirements
- Components/containers can also
  - Communicate via a *middleware bus* &
  - Reuse *common middleware services*

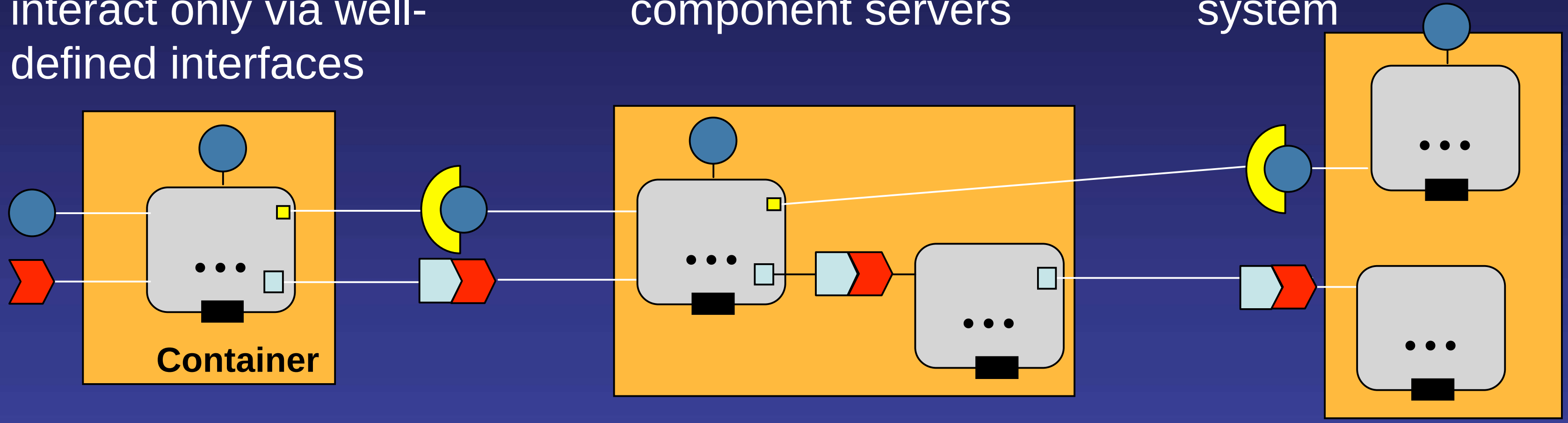
Component middleware defines interfaces, policies, & *some* implementations

Component Middleware

- Creates a standard “virtual boundary” around application **component** implementations that interact only via well-defined interfaces

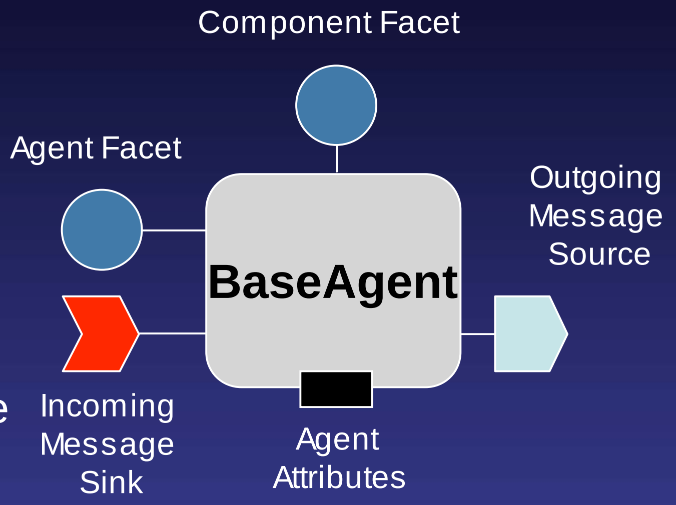
- Define standard **container** mechanisms needed to execute components in generic component servers

- Specify the infrastructure needed to **configure & deploy** components throughout a distributed system



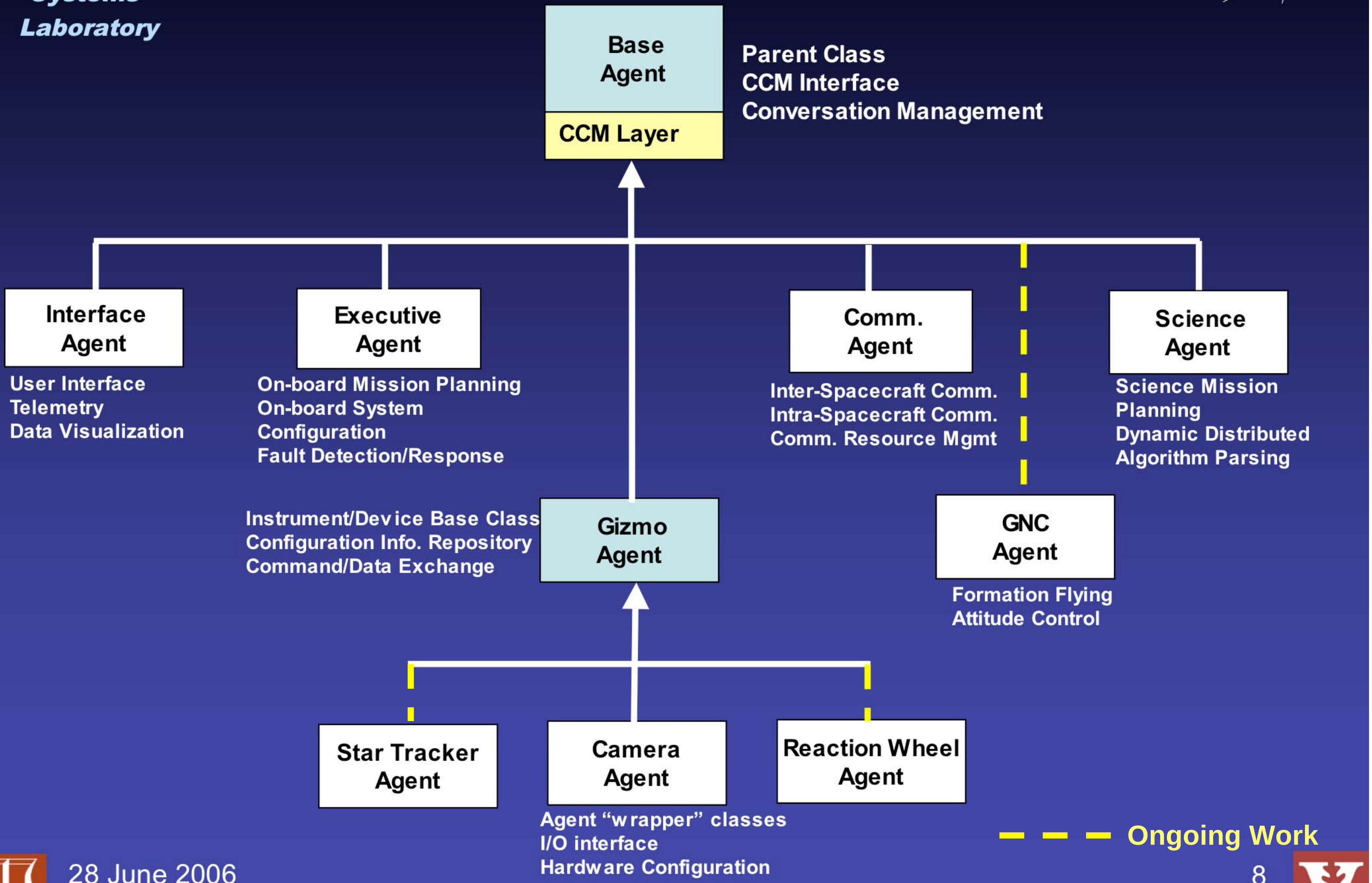
```
<ComponentAssemblyDescription id="a_HUDDisplay"> ...
 <connection>
   <name>GPS-RateGen</name>
   <internalEndPoint><portName>Refresh</portName><instance>a_GPS</instance></internalEndPoint>
   <internalEndPoint><portName>Pulse</portName><instance>a_RateGen</instance></internalEndPoint>
  </connection>
  <connection>
    <name>NavDisplay-GPS</name>
    <internalEndPoint><portName>Refresh</portName><instance>a_NavDisplay</instance></internalEndPoint>
    <internalEndPoint><portName>Ready</portName><instance>a_GPS</instance></internalEndPoint>
  </connection> ...
</ComponentAssemblyDescription>
```

ANA Agent as a CORBA Component



- All agents have a standard set of capabilities captured by a BaseAgent CORBA Component
- All agents inherit ports and some implementation code from BaseAgent
  - Agent Facet provides access to limited agent management needed by Executive and Interface agents (i.e. name, state, heartbeat rate, wakeup/doze commands)
  - Outgoing Message event source for transmission of messages
  - Incoming Message event sink for reception of messages
  - Attributes for agent name and type (e.g Science Agent)
- Component member functions delegate to derived agent code (using Bridge Pattern) for agent-specific implementations
- Naming and RT Event services have been integrated into CIAO for agent discovery and message transport

ANA Class Hierarchy



Base Agent

CCM Layer

Parent Class
CCM Interface
Conversation Management

Interface Agent

User Interface
Telemetry
Data Visualization

Executive Agent

On-board Mission Planning
On-board System Configuration
Fault Detection/Response

Comm. Agent

Inter-Spacecraft Comm.
Intra-Spacecraft Comm.
Comm. Resource Mgmt

Science Agent

Science Mission Planning
Dynamic Distributed Algorithm Parsing

Instrument/Device Base Class
Configuration Info. Repository
Command/Data Exchange

Gizmo Agent

GNC Agent

Formation Flying
Attitude Control

Star Tracker Agent

Camera Agent

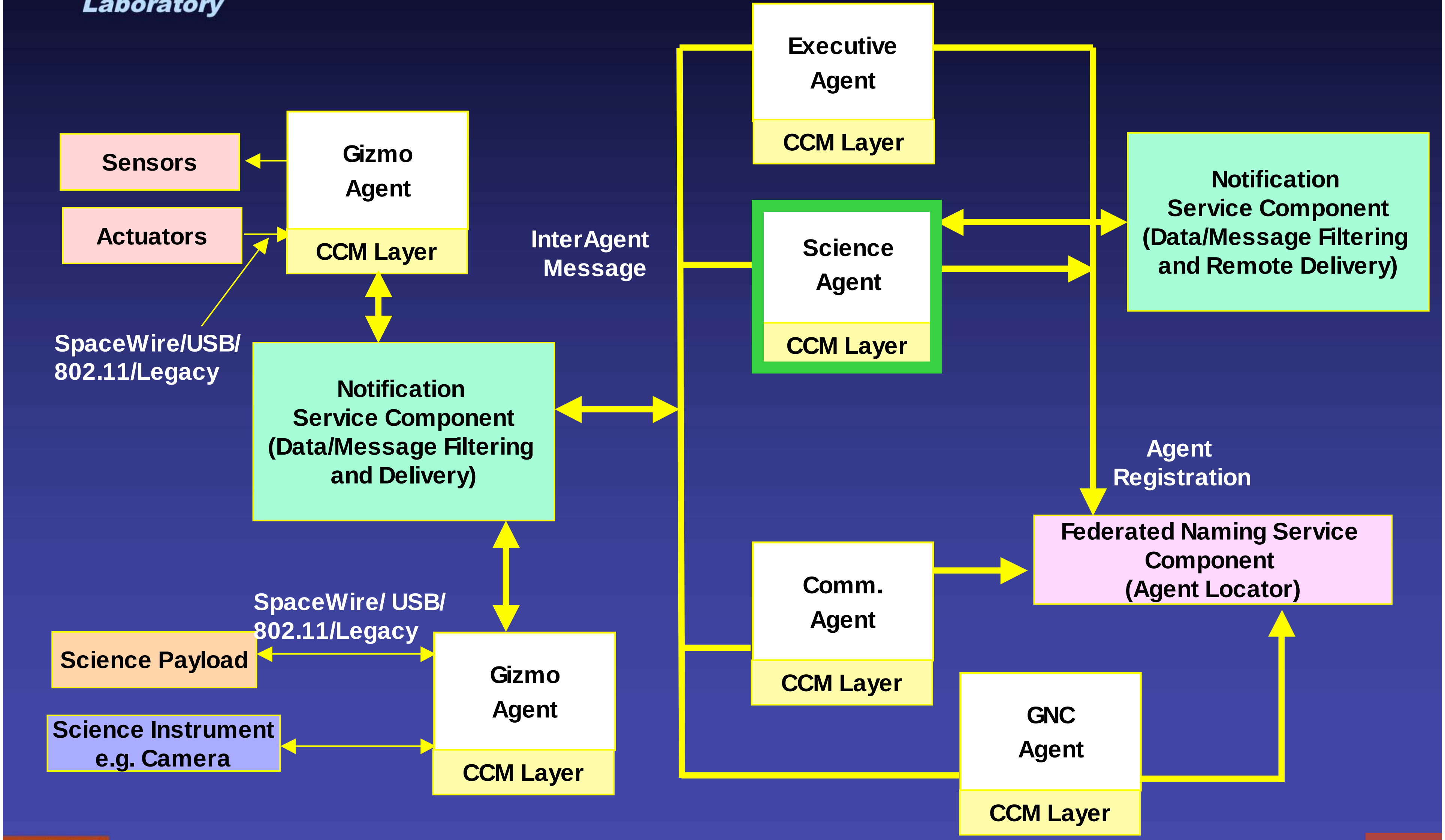
Agent “wrapper” classes
I/O interface
Hardware Configuration

Reaction Wheel Agent

Ongoing Work

DSL
Distributed
Systems
Laboratory

Example ANA Deployment for a Single Host

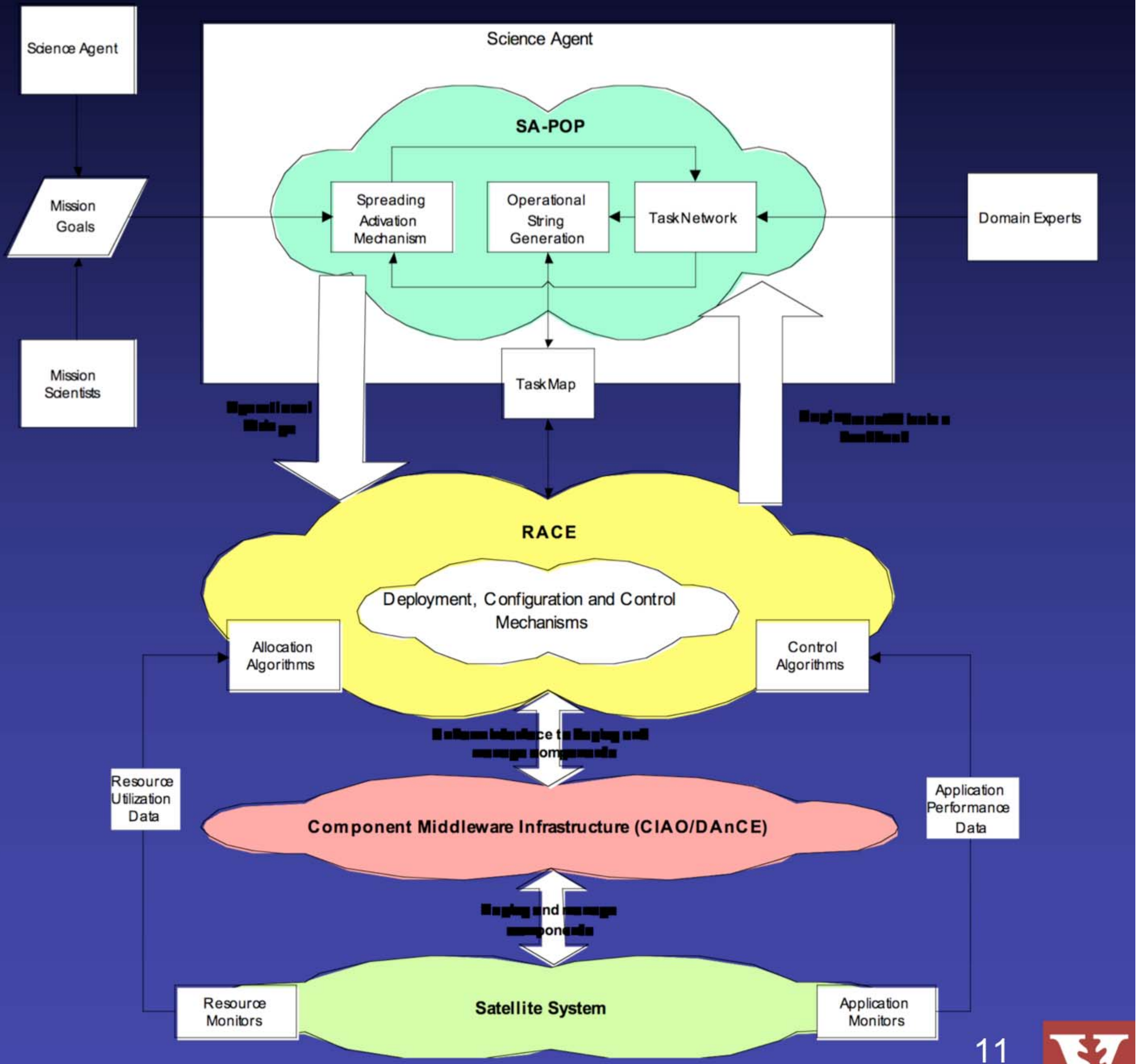


Science Agent Responsibilities

- Meet the science mission goals by managing payload sensors and onboard data processing
- Range of tasks involve
  - Data Collection
    - Operational mode changes
    - Sensor selection and reconfiguration
  - Data Analysis and Processing using a suite of parameterized algorithms
    - Data compression
    - Change detection
  - Limited inter-spacecraft communication associated with science mission
    - Generate telemetry stream
    - Coordinated observations and data sharing

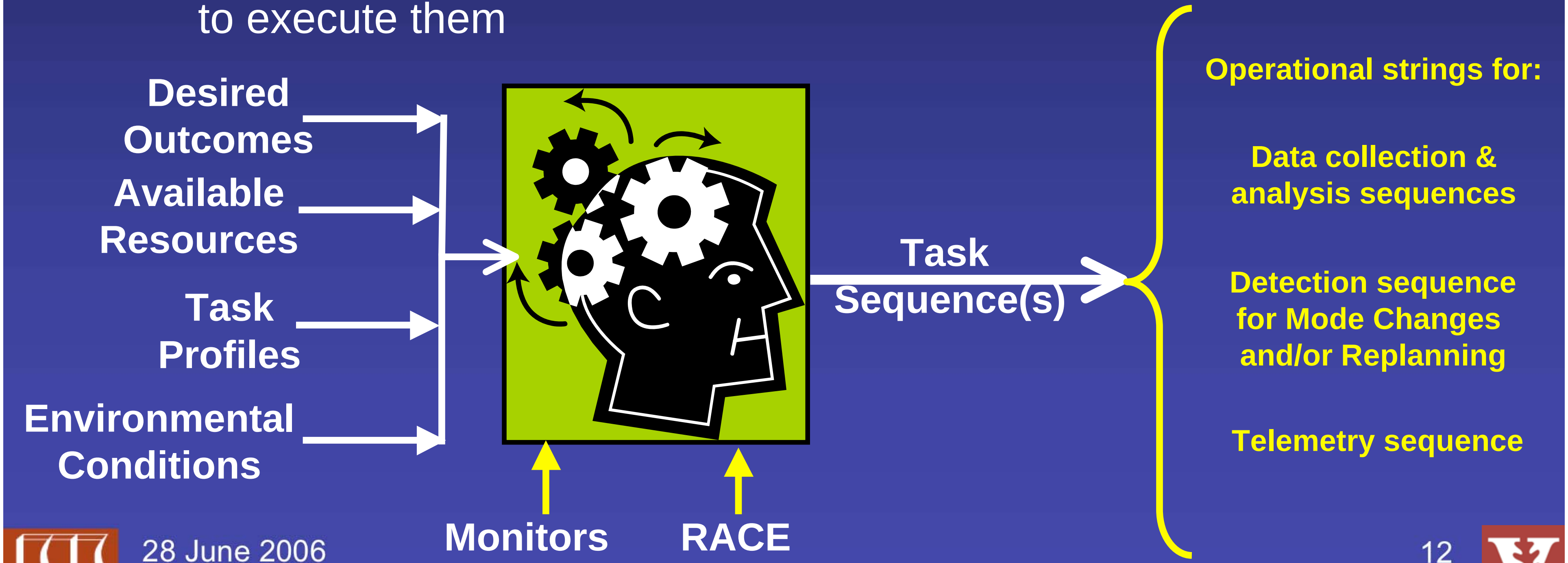
Science Agent Realization

- Manages the onboard data processing required to achieve science mission goals
- Operational Strings
  - Representation of application dependent workflow sequences
  - Serial/parallel sequences of tasks with temporal constraints
- Onboard processing main subsystems
  - Spreading Activation Partial Order Planner (SA-POP)
  - Resource Allocation and Control Engine (RACE)


Science Agent
Mission Goals
Mission Scientists
Science Agent
SA-POP
Spreading Activation Mechanism
Operational String Generation
Task Network
Domain Experts
Task Map
RACE
Deployment, Configuration and Control Mechanisms
Allocation Algorithms
Control Algorithms
Resource Utilization Data
Application Performance Data
Component Middleware Infrastructure (CIAO/DAnCE)
Satellite System
Resource Monitors
Application Monitors

SA-PO Planner – Objectives

- Generate desired outcomes or meet goals
  - Generate right task sequence
  - Tasks need to be properly parameterized
  - Parameters of task sequence must be matched or consistent with each other
  - Task parameters chosen such that sufficient resources available to execute them



Desired Outcomes

Available Resources

Task Profiles

Environmental Conditions

Task Sequence(s)

Monitors

RACE

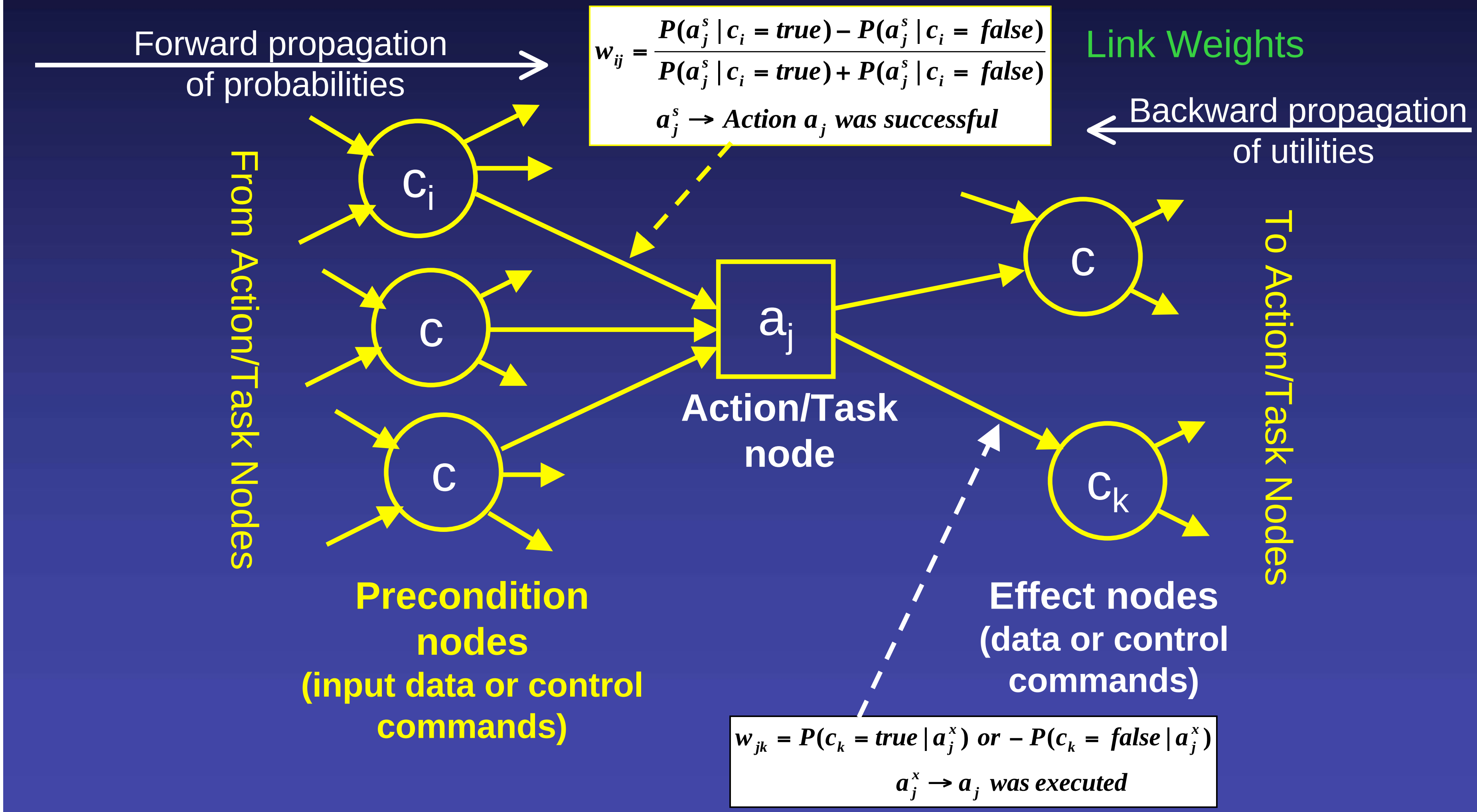
Operational strings for:

Data collection & analysis sequences

Detection sequence for Mode Changes and/or Replanning

Telemetry sequence

Dynamic Task Planning using Spreading Activation



Forward propagation of probabilities

$$w_{ij} = \frac{P(a_j^s \mid c_i = true) - P(a_j^s \mid c_i = false)}{P(a_j^s \mid c_i = true) + P(a_j^s \mid c_i = false)}$$

$a_j^s \rightarrow Action\ a_j\ was\ successful$

Link Weights

Backward propagation of utilities

Action/Task node

Precondition nodes (input data or control commands)

Effect nodes (data or control commands)

$w_{jk} = P(c_k = true \mid a_j^x)\ or\ - P(c_k = false \mid a_j^x)$

$a_j^x \rightarrow a_j\ was\ executed$

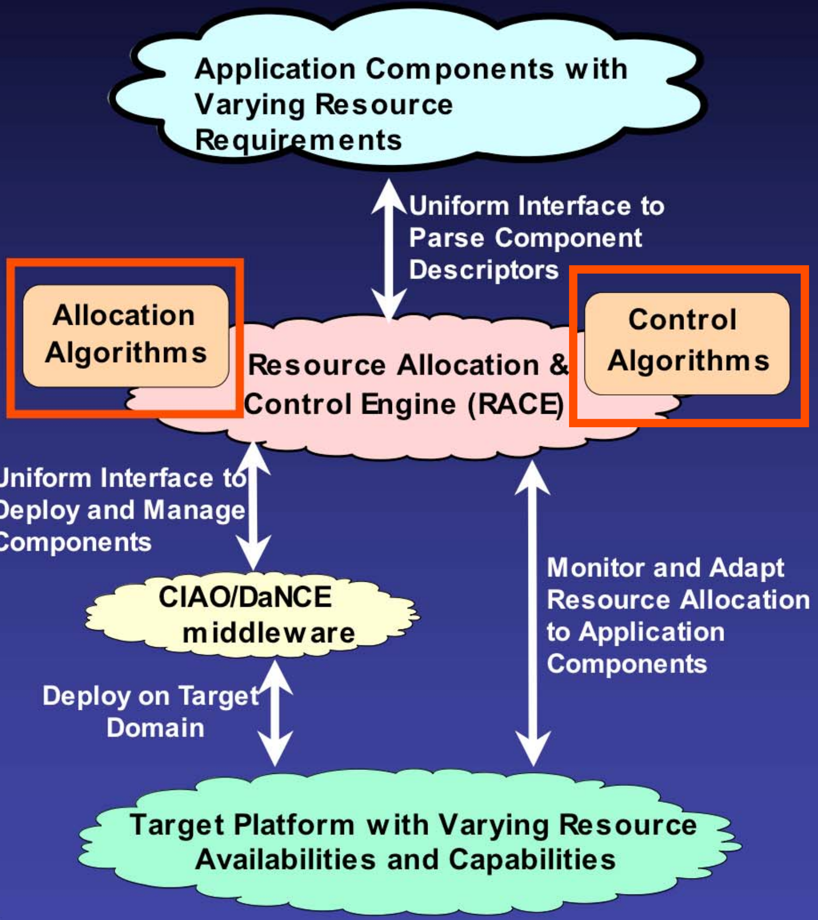
Advantages of Spreading Activation Partial Order Planner

- Has the *anytime property*.
  - More spreading activation "considers" longer plans,
  - But action selection can occur at any time.
- Balances reactive action selection with deliberative expected utility computation.
- Takes advantage of fortuitous events as they occur.

DSL Distributed Systems Laboratory

Resource Allocation and Control Engine (RACE)

- Dynamic resource management framework atop CORBA Component Model (CCM) middleware (CIAO/DAnCE)
- Allocates components to available resources
- Configure components to satisfy QoS requirements based on dynamic mission goals
- Perform run-time adaptation
  - *Coarse-grained adaptation mechanisms*
    - React to new missions, drastic changes in mission goals, or unexpected circumstances such as loss of resources
    - e.g., component re-allocation or migration
  - *Fine-grained adaptation mechanisms*
    - Compensate for drift & smaller variations in resource usage
    - e.g., adjustment of application parameters, such as QoS settings



Application Components with Varying Resource Requirements
Uniform Interface to Parse Component Descriptors
Allocation Algorithms
Resource Allocation & Control Engine (RACE)
Control Algorithms
Uniform Interface to Deploy and Manage Components
CIAO/DaNCE middleware
Monitor and Adapt Resource Allocation to Application Components
Deploy on Target Domain
Target Platform with Varying Resource Availabilities and Capabilities

Magnetospheric Multi-Scale (MMS) Mission

- Mission Goal

  "To study the microphysics of three fundamental plasma processes: magnetic reconnection, energetic particle acceleration, and turbulence*"

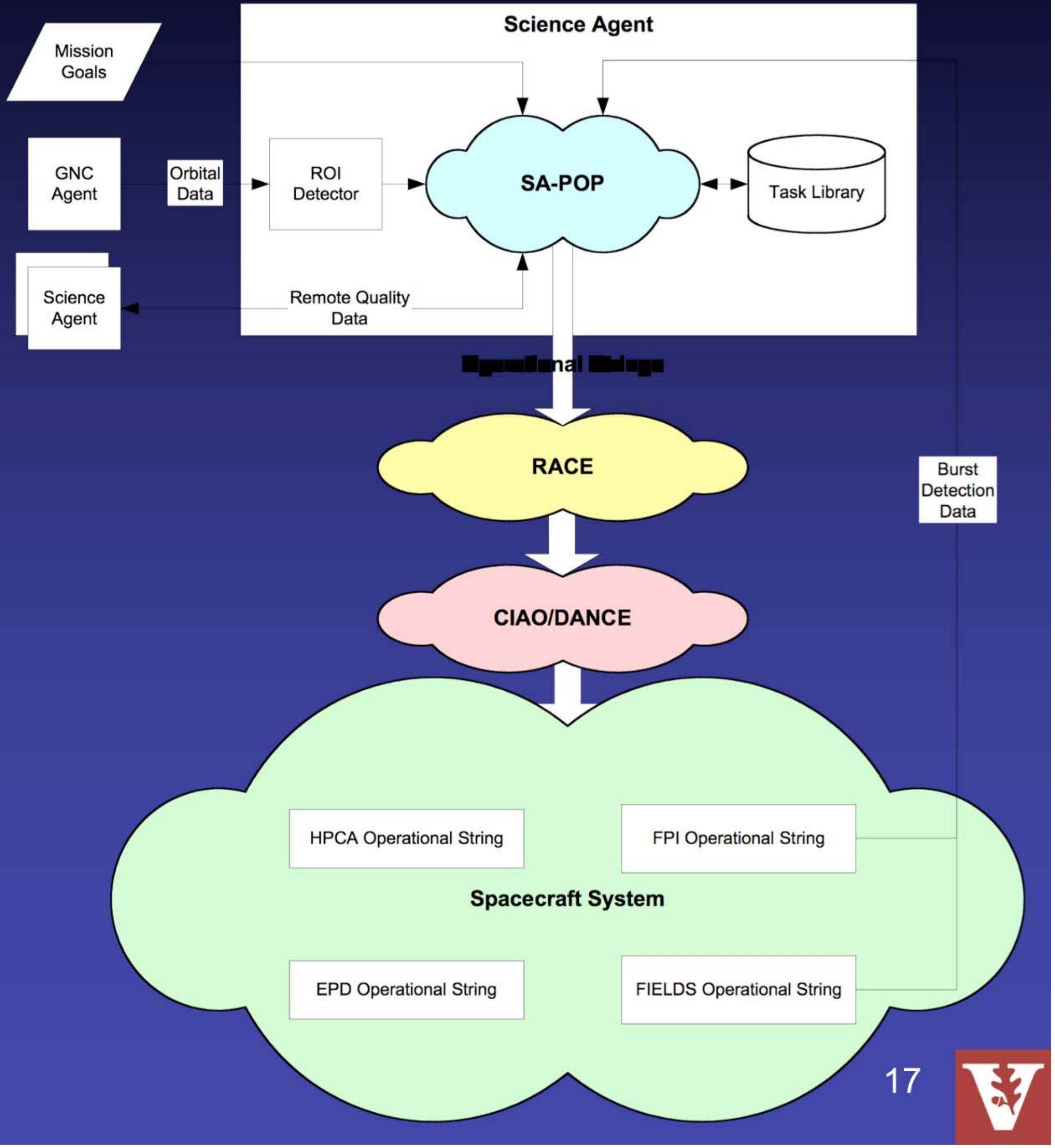
- Architecture
  - Constellation of 4 identical spacecraft in a tetrahedral format with a suite of 4 primary sensors/subsystems
  - Three data collection modes: slow, fast, and burst. Each mode includes different goals, orbits, and data priorities.
- Dynamic Environment
  - Changing goal prescriptions
    - Specified by mission scientists
    - Autonomous mode changes driven by satellite positions and the results of analyzing collected data
  - Changing system conditions (e.g. resource usage) and environmental conditions (e.g. transient electromagnetic events of interest)

* SWRI MMS webpage http://mms.space.swri.edu/

DSL
Distributed
Systems
Laboratory

Science Agent for MMS

- SA-POP in Science Agent generates / modifies active operational strings for each instrument based on
  - Mission goals from scientists
  - Orbital location
  - Local and remote quality of sensor measurements
- Task library contains algorithms for onboard processing
  - Lossless/lossy compression
  - Higher order data fusion for burst detection
- RACE (re)deploys components on available processing nodes based on plan given by Science Agent / SA-POP and resource utilization



Conclusions and Future Work

- The ANA shows promise of reducing cost and technical complexity of autonomous missions that are heterogeneous both in objectives and configuration i.e. Agent technology is an appropriate paradigm for design and implementation
- Leveraging mature terrestrial standards as a foundation is prudent and ensures that the application is robust and the development effort is well managed
- Science Agent functional decomposition between the spreading activation partial order planner and RACE has several benefits in terms of
  - Flexibility
    - Pluggable allocation and control algorithms in RACE for different domains.
    - Shared task map allows substitution of functionally equivalent task implementations by RACE
  - Scalability
    - Separation of concerns between SA-POP and RACE limits search spaces in each.
    - SA-POP handles cascading planning choices in operational string generation, but only considers resource allocation *feasibility* with *coarse-grained* (summed across processing nodes) resource constraints.
    - RACE handles resource allocation *optimization* with *fine-grained* (individual processing nodes) resource constraints and dynamic control, but for a fixed operational string.
  - Dynamism
    - Anytime algorithm for planning/replanning achieved on-board satellites with incrementally updated task network and (usually) fast plan repair.
    - Reconfiguration and (re)deployment of CCM Components by RACE to meet resource constraints under uncertain conditions
- ANA is continuing development under Internal R&D funding, and potentially under future AIST ROSES program…..

Acronyms

ACE	ADAPTIVE Communications Environment
ANA	Adaptive Network Architecture
CCM	CORBA Component Model
CIAO	Component Integrated ACE ORB
CORBA	Common Object Request Broker Architecture
CoSMIC	Component Synthesis with Model Integrated Computing
D & C	Deployment and Configuration
DAnCE	Deployment and Configuration Engine
DRE	Distributed Real-Time Embedded
EPD	Energetic Particle Detector
FCE	Formation Computing Environment
FPI	Fast Plasma Instrument
FIELDS	Electromagnetic Field Instruments
GME	Generic Modeling Environment

HPCA	Hot Plasma Composition Analyzer
IRAS	Inter-Satellite Ranging and Alarm System
MDD	Model Driven Design
MMS	Magnetospheric MultiScale
OCL	Object Constraint Language
OMG	Object Management Group
ORB	Object Request Broker
PICML	Platform Independent Component Modeling Language
PIM	Platform Independent Model
PSM	Platform Specific Model
RACE	Resource Allocation and Control Engine
ROI	Region of Interest
SA-POP	Spread Activation – Partial Order Planner
TAO	The ACE ORB